AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously presented) A method in a data processing system containing source code with a subprogram having at least one of an integer non-scalar parameter and a logical non-scalar parameter, the method comprising:

creating an interface file for the subprogram in the source code;

storing in the interface file a definition of the subprogram;

adding to the interface file a directionality of at least one of the integer parameter and the logical parameter based on comments in the source code;

adding to the interface file a parameter size along each dimension of at least one of the integer parameter and the logical parameter; and

reading the interface file to generate a stub routine that converts at least one of the integer and logical parameters from 32-bit to 64-bit and that invokes the subprogram by specifying the converted parameters.

2. (Original) The method of claim 1, wherein the source code is 32-bit code and wherein the method further includes the step of:

invoking the 64-bit code from 32-bit code.

3. (Currently amended) A method in a data processing system, comprising ~~the steps of~~:

receiving 32-bit source code;

generating, from the 32-bit source code, a 32-bit interface file including statements describing characteristics of parameters in the 32-bit source code; and

automatically generating, based on the statements in the 32-bit interface file, a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code.

4. (Currently amended) The method of claim 3, wherein the 32-bit source code ~~has a subprogram with~~ includes at least one of an integer parameter and a ~~or~~ logical parameter and wherein ~~the automatically~~ generating ~~step further~~ a 32-bit interface includes ~~the steps of~~:

~~creating an interface for the subprogram;~~

~~inserting a code generator statement into the interface describing a characteristic of the parameter;~~

determining whether the at least one of an integer and logical parameter has input directionality, output directionality, or input and output directionality; and

inserting into the 32-bit interface file code generator statements corresponding to the determined directionality of the at least one parameter

~~using the interface to create a stub for use as a 32-bit to 64-bit converter~~.

5. (Currently amended) A data processing system, comprising:

a storage device, comprising:

source code with a subprogram having at least one of an integer and logical parameter;

an interface generator that reads the subprogram and that generates an interface file with indications of characteristics of the parameter; and

a stub generator that reads the interface file and that generates a stub for the subprogram by using the characteristics, wherein ~~each of~~ the ~~stubs~~ stub receives a set of parameter values, generates the values for the required parameters from the received set of parameter values, and invokes the subprogram with the values for the parameters; and

a processor for running the interface generator and the stub generator.

6. (Original) The data processing system of claim 5, wherein the source code contains comments indicating the characteristics of the parameter.

7. (Original) The data processing system of claim 6, wherein the characteristics include an indication of a conditional value for at least one of the required parameters.

8. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether at least one of the required parameters is used to contain a return value.

9. (Original) The data processing system of claim 6, wherein the characteristics include a directionality of at least one of the required parameters.

10. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether at least one of the required parameters returns a multidimensional variable.

11. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether a size of at least one of the required parameters is based on another one of the required parameters.

12. (Original) The data processing system of claim 6, wherein the characteristics include an indication of whether at least one of the required parameters is a work space parameter.

13. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method comprising the steps of:

receiving 32-bit source code;

generating, from the 32-bit source code, a 32-bit interface file including statements describing characteristics of parameters in the 32-bit source code; and

automatically generating a 32-bit interface to 64-bit source code based on the statements in the interface file.

14. (Currently amended) The computer-readable medium of claim 13, wherein the 32-bit source code has a subprogram with a parameter and wherein ~~the automatically~~ generating ~~step further~~ a 32-bit interface file includes ~~the steps of~~:

~~creating an interface for the subprogram;~~

~~inserting a code generator statement into the interface describing a characteristic of the parameter;~~

determining whether the parameter in the subprogram has input directionality, output directionality, or input and output directionality; and

inserting into the 32-bit interface file code generator statements corresponding to the determined directionality of the parameter in the subprogram

~~using the interface to create a stub for use as the interface to the 64-bit code~~.

15. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having source code with a subprogram having a parameter, the method comprising the steps of:

reading the source code;

generating from the source code an interface file including characteristics of the parameter; and

generating, based on the characteristics of the parameter, a stub routine that invokes the subprogram and that facilitates use of at least one of a converted integer and logical parameter.

16. (Currently amended) A data processing system comprising:

means for receiving 32-bit source code;

means for generating, from the 32-bit source code, a 32-bit interface file including statements describing characteristics of parameters in the 32-bit source code; and

means for automatically generating, based on the statements in the 32-bit interface file, a 32-bit to 64-bit conversion stub that is used by the 32-bit source code to invoke 64-bit code.

17. (New) The method of claim 1, wherein adding to the interface file a directionality includes:

determining whether the at least one parameter has input directionality, output directionality, or input and output directionality;

adding to the interface file statements based on the determined directionality.

18. (New) The method of claim 1, wherein adding to the interface file a parameter size includes:

adding to the interface file statements indicating a number of dimensions of the at least one parameter and a number of elements in each dimension.

19. (New) The method of claim 3, wherein generating a 32-bit interface file includes invoking an interface generator that:

scans the 32-bit source code and creates the interface file according to a definition; and

adds to the interface file the statements describing characteristics of the parameters by parsing the 32-bit source code.

20. (New) The method of claim 19, wherein automatically generating a 32-bit to 64-bit conversion stub includes invoking a stub generator that:

reads the 32-bit interface file to populate a hash table with information identifying the parameters in the interface file;

re-reads the 32-bit interface file to populate the hash table with information indicating processing that occurs for the statements in the interface file; and

generates the 32-bit to 64-bit conversion stub using the hash table.